

MMM	MMM	AAAAAAAAA	CCCCCCCCCCCCC	RRRRRRRRRRRRR	000000000			
MMM	MMM	AAAAAAAAA	CCCCCCCCCCCCC	RRRRRRRRRRRRR	000000000			
MMM	MMM	AAAAAAAAA	CCCCCCCCCCCCC	RRRRRRRRRRRRR	000000000			
MMMMMM	MMMMMM	AAA	AAA	CCC	RRR	RRR	000	000
MMMMMM	MMMMMM	AAA	AAA	CCC	RRR	RRR	000	000
MMMMMM	MMMMMM	AAA	AAA	CCC	RRR	RRR	000	000
MMM	MMM	AAA	AAA	CCC	RRR	RRR	000	000
MMM	MMM	AAA	AAA	CCC	RRR	RRR	000	000
MMM	MMM	AAA	AAA	CCC	RRR	RRR	000	000
MMM	MMM	AAA	AAA	CCC	RRRRRRRRRRRRR		000	000
MMM	MMM	AAA	AAA	CCC	RRRRRRRRRRRRR		000	000
MMM	MMM	AAA	AAA	CCC	RRRRRRRRRRRRR		000	000
MMM	MMM	AAAAAAAAAAAAAAAAA	CCC	RRR	RRR		000	000
MMM	MMM	AAAAAAAAAAAAAAAAA	CCC	RRR	RRR		000	000
MMM	MMM	AAAAAAAAAAAAAAAAA	CCC	RRR	RRR		000	000
MMM	MMM	AAA	AAA	CCC	RRR	RRR	000	000
MMM	MMM	AAA	AAA	CCC	RRR	RRR	000	000
MMM	MMM	AAA	AAA	CCC	RRR	RRR	000	000
MMM	MMM	AAA	AAA	CCC	RRR	RRR	000	000
MMM	MMM	AAA	AAA	CCCCCCCCCCCCC	RRR	RRR	000000000	
MMM	MMM	AAA	AAA	CCCCCCCCCCCCC	RRR	RRR	000000000	
MMM	MMM	AAA	AAA	CCCCCCCCCCCCC	RRR	RRR	000000000	

```

LL               IIIII
LL             IIIIII
LL           II
LL         I
LL       I
LL     I
LL   I
LL I
LL
SSSSSSS
SSSSSSS
SS
SS
SS
SS
SSSSSS
SSSSSS
SS
SS
SS
SS
SSSSSSS
SSSSSSS

```

[illegible]

(2)	62	DECLARATIONS
(3)	86	MACRO DEFINE A MACRO
(4)	154	DELETE .MDELETE DIRECTIVE
(5)	190	FREE UP MACRO DELETE OLD MACRO DEFINITION
(6)	245	MACRO_DEFINE CREATE MNB AND SET IT UP
(7)	284	MACRO_ARG_SCAN SCAN FORMAL MACRO ARGUMENTS



```
0000 1      .TITLE  MAC$MACDEF      MACRO DEFINITION PROCESSOR
0000 2      .IDENT  'V04-000'
0000 3
0000 4
0000 5      *****
0000 6      *
0000 7      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      *  ALL RIGHTS RESERVED.
0000 10     *
0000 11     *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12     *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13     *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14     *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15     *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16     *  TRANSFERRED.
0000 17     *
0000 18     *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19     *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20     *  CORPORATION.
0000 21     *
0000 22     *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23     *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24     *
0000 25     *
0000 26     *****
0000 27
0000 28
0000 29     ++
0000 30     FACILITY:      VAX MACRO ASSEMBLER OBJECT LIBRARY
0000 31
0000 32     ABSTRACT:
0000 33
0000 34     The VAX-11 MACRO assembler translates MACRO-32 source code into object
0000 35     modules for input to the VAX-11 LINKER.
0000 36
0000 37     ENVIRONMENT:  USER MODE
0000 38
0000 39     AUTHOR: Benn Schreiber, CREATION DATE: 20-AUG-78
0000 40
0000 41     MODIFIED BY:
0000 42
0000 43     V03.00  MTR0009      Mike Rhodes      15-Mar-1982
0000 44             Modify routine MACRO:: to de-optimize the handling of
0000 45             null macros. Fixes SPR #11-42119.
0000 46
0000 47     V01.10  RN0023      R. Newland      3-Nov-1979
0000 48             New message codes to get error messages from system
0000 49             message file.
0000 50
0000 51     V01.09  RN0010      R. Newland      5-Sep-1979
0000 52             Multipage MXB blocks
0000 53
0000 54     V01.08  RN0008      R. Newland      31-Aug-1979
0000 55             31 character symbols (macro names)
0000 56
0000 57     V01.07  RN0005      R. Newland      13-Aug-1979
```

MACSMACDEF  
V04-000

MACRO DEFINITION PROCESSOR

C 6

16-SEP-1984 02:07:14  
5-SEP-1984 01:48:54

VAX/VMS Macro V04-00  
[MACRO.SRC]MACDEF.MAR;1

Page 2  
(1)

0000 58 :  
0000 59 :  
0000 60 :--

Variable symbol name storage

MA  
Ps  
SA  
MA  
  
Ph  
--  
In  
Co  
Pa  
Sy  
Pa  
Sy  
Ps  
Cr  
As  
  
Th  
31  
Th  
43  
16  
  
Ma  
--  
-S  
-S  
TO  
58  
Th  
MA

```
0000 62      .SBTTL  DECLARATIONS
0000 63      :
0000 64      : INCLUDE FILES:
0000 65      :
0000 66      :
0000 67      :
0000 68      : MACROS:
0000 69      :
0000 70      :
0000 71      $MAC_SYMBLKDEF      ;DEFINE SYMBOL BLOCK OFFSETS
0000 72      $MAC_INTCODDEF      ;DEFINE INT. FILE CODES
0000 73      $MAC_INPBLKDEF      ;DEFINE INPUT BLOCK OFFSETS
003C 74      $MAC_MNBDEF         ;DEFINE MNB/MAB OFFSETS
003C 75      $MAC_GENVALDEF      ;DEFINE GENERAL VALUES
003C 76      $MAC_CRFLAGDEF      ;DEFINE CREF CONTROL FLAGS
003C 77      $MACMSGDEF          ; Define message codes
003C 78      $MAC_CTLFLGDEF      ;DEFINE CONTROL FLAGS
003C 79      :
003C 80      :
003C 81      : EQUATED SYMBOLS:
003C 82      :
003C 83      :
00000000 84      .PSECT  MAC$RO_CODE_MAC,NOWRT,GBL, LONG
```



```
0000 86 .SBTTL MACRO DEFINE A MACRO
0000 87
0000 88 :++
0000 89 : FUNCTIONAL DESCRIPTION:
0000 90 :
0000 91 : THIS ROUTINE IS CALLED WHEN A '.MACRO' DIRECTIVE IS SCANNED.
0000 92 : THE MACRO FORMAL ARGS ARE SCANNED, AND THE MACRO BODY IS
0000 93 : SCANNED AND PLACED IN CORE. THE MACRO DEFINITION IS ENTERED
0000 94 : IN THE USER MACRO HASH TABLE.
0000 95 :
0000 96 :--
0000 97
0000 98 MACRO::
0000 99 :DIRECTIVE = KMACRO
OE 0005'CF E8 0000 100 BLBS W^LST$G_MACRODEF+SYMSL VAL,10$ :BRANCH IF LISTING MACRO DEFINITIONS
0005 101 $INTOUT_LW INT$-SETLONG,<#0,#MAC$GL LIST IT> :NO--DO NOT LIST
FFEA' 30 0013 101 10$: BSBW MAC$SYMSCNUP :SCAN MACRO NAME
OB 50 E8 0016 102 BLBS RO,20$ :BRANCH IF NAME FOUND
0019 103 $MAC_ERR ILLMACNAM : No--get message code
5A OD D0 001E 104 MOVL #CR,R10 :FORCE NEW LINE
FFDC' 31 0021 105 BRW MAC$ERRORLN :ISSUE ERROR AND RETURN
53 0000'CF 9E 0024 106 20$: MOVAB W^MAC$AL_UMCHSHTB,R3 :POINT TO USER MACRO HASH TABLE
FFD4' 30 0029 107 BSBW MAC$SRCSYMTAB :LOOK UP THE MACRO NAME
OB 50 E9 002C 108 BLBC RO,30$ :BRANCH IF NOT FOUND
52 DD 002F 109 PUSHL R2 :FOUND--SAVE PTR TO PREVIOUS
00BE 30 0031 110 BSBW FREE_UP_MACRO :DELETE OLD DEFINITION
52 8ED0 0034 111 POPL R2 :RESTORE PTR TO PREVIOUS
0126 30 0037 112 30$: BSBW MACRO DEFINE :DEFINE THE MACRO
55 00'8F 9A 003A 113 MOVZBL #CRFSR DEF,R5 :THIS IS MACRO DEFINITION
FFBF' 30 003E 114 BSBW MAC$CREF_MACRO :CREF IT IF CREFFING
FFBC' 30 0041 115 BSBW MAC$SKIPSP :SKIP SPACES
2C 5A 91 0044 116 CMPB R10,#^A/,/ :STOP ON A COMMA?
03 12 0047 117 BNEQ 40$ :IF NEQ NO
FFB4' 30 0049 118 BSBW MAC$GETCHR :YES--SKIP IT
004C 119 :
004C 120 : SCAN FORMAL ARGUMENTS
004C 121 :
004C 122 40$: BSBW MACRO_ARG_SCAN :SCAN FORMAL ARGUMENTS AND SET UP
0161 30 004F 123 :MAB'S FOR THEM
05 A6 0000'CF D0 004F 124 MOVL W^MAC$GL_ARGPTR,MNBSL_TXTIP(R6) :STORE POINTER TO TEXT
5A OD 9A 0055 125 MOVZBL #CR,R10 :FORCE READING OF NEW LINE
FFA5' 30 0058 126 BSBW MAC$GETCHR :READ FIRST CHARACTER OF BODY
18 A6 DD 005B 127 PUSHL MNBSL_ARGP(R6) :STACK THE ARGUMENT POINTER
0000'CF C3 005E 128 SUBL3 W^MAC$GL_BLKPTR,- :FIGURE SPACE LEFT IN BLOCK
50 0000'CF 0062 129 W^MAC$GL_ARGPTR,RO : (THIS GIVES SPACE USED)
7E 00000200 8F 50 C3 0066 130 SUBL3 RO,#512,=(SP) :FIGURE SPACE LEFT AND STACK IT
0000'CF DD 006E 131 PUSHL W^MAC$GL_ARGPTR :STACK POINTER TO FREE SPACE IN BLOCK
0000'CF 03 FB 0072 132 CALLS #3,W^MAC$BODY_SCAN :SCAN THE MACRO BODY AND READ
0077 133 :IT INTO CORE
0077 134 :
0077 135 : MACRO IS SCANNED. SEE IF THERE WAS A SYMBOL NAME FOLLOWING THE
0077 136 : .ENDM DIRECTIVE.
0077 137 :
0000'CF D6 0077 138 50$: INCL W^MAC$GL MCDEF :COUNT ANOTHER MACRO DEFINITION
FF82' 30 007B 139 BSBW MAC$SYMSCNUP :IS THERE A SYMBOL?
2D 50 E9 007E 140 BLBC RO,60$ :IF LBC NO
50 0000'CF D0 0081 141 MOVL W^MAC$GL_MACPTR,RO :YES--GET PTR TO MNB OF MACRO
51 04 A0 9A 0086 142 MOVZBL MNBSB_NAME(RO),R1 : Get offset to symbol count/name
```

62	53	00	50	51	C2	008A	143	SUBL2	R1,R0	:	and form its address
			51	80	9A	008D	144	MOVZBL	(R0)+,R1	:	GET SIZE OF NAME
			0000	CF	9E	0090	145	MOVAB	W*MAC\$AB-TMP\$YM,R2	:	POINT TO SCANNED NAME
			53	82	9A	0095	146	MOVZBL	(R2)+,R3	:	GET ITS SIZE
			60	51	2D	0098	147	CMPC5	R1,(R0),#0,R3,(R2)	:	RIGHT .ENDM FOR THIS MACRO?
				0E	13	009E	148	BEQL	60\$	:	IF EQL YES
						00A0	149	\$INTOUT	X INT\$ CHKL	:	NO--ALIGN SOURCE AND LISTING
						00A6	150	\$MAC_ERR	ENDWRNGMAC	:	Get message code
			FF52		31	00AB	151	BRW	MAC\$ERRORPT	:	ISSUE AND RETURN
					05	00AE	152	RSB		:	

60\$:



```
00AF 154 .SBTTL DELETE .MDELETE DIRECTIVE
00AF 155
00AF 156 :++
00AF 157 : FUNCTIONAL DESCRIPTION:
00AF 158 :
00AF 159 : THIS ROUTINE IS CALLED TO PROCESS THE .MDELETE DIRECTIVE. THE
00AF 160 : ARGUMENT TO THE DIRECTIVE IS A LIST OF MACRO NAMES. IF ANY
00AF 161 : OF THE MACROS NAMED ARE DEFINED, THEY ARE UNDEFINED. THE LITERAL
00AF 162 : PRINTED WITH THE LISTING IS THE NUMBER OF MACROS DELETED.
00AF 163 :
00AF 164 :--
00AF 165
00AF 166 DELETE::
53 7E D4 00AF 167 CLRL -(SP) ;DIRECTIVE = KMDELETE
FF4C' 30 00B1 168 10$: BSBW MAC$SYMSCNUP ;INITIALIZE COUNT OF MACROS DELETED
OD 50 E8 00B4 169 BLBS RO,20$ ;SCAN A MACRO NAME
FF41' 30 00B7 170 $MAC_ERR DIRSYN ;BRANCH IF WE FOUND ONE
5A OD D0 00BC 171 BSBW- MAC$ERRORPT ; No-directive syntax error
22 11 00BF 172 MOVL #CR,R10 ;REPORT ERROR
0000'CF 9E 00C2 173 BRB 50$ ;FORCE NEW LINE
FF34' 30 00C4 174 20$: MOVAB W^MAC$AL UMCHSHTB,R3 ;FINISH PROCESSING
04 50 E9 00C9 175 BSBW MAC$SRCSYMTAB ;LOOK UP IN MACRO HASH TABLE
21 10 00CC 176 BLBC RO,30$ ;BRANCH IF NOT FOUND (EASY OUT)
6E D6 00CF 177 BSBB FREE_UP_MACRO ;FOUND--DELETE DEFINITION
FF2A' 30 00D1 178 INCL (SP) ;COUNT MACRO AS DELETED
2C 5A 91 00D3 179 30$: BSBW MAC$SKIPSP ;SKIP SPACES
06 12 00D9 180 CMPB R10,#^A/,/ ;STOP ON A COMMA?
FF22' 30 00DB 181 BNEQ 40$ ;IF NEQ NO
FF1F' 30 00DE 182 BSBW MAC$GETCHR ;YES--SKIP OVER IT
OD 5A 91 00E1 183 BSBW MAC$SKIPSP ;SKIP SPACES AGAIN
CB 12 00E4 184 40$: CMPB R10,#CR ;AT EOL?
53 8ED0 00E6 185 50$: BNEQ 10$ ;IF NEQ NO--KEEP SCANNING
00E9 186 POPL R3 ;YES--GET COUNT OF MACROS DELETED
00F1 187 $INTOUT_LW INT$_PRIL,R3 ;PRINT # MACROS DELETED
05 00F1 188 RSB
```

```
00F2 190 .SBTTL FREE_UP_MACRO DELETE OLD MACRO DEFINITION
00F2 191
00F2 192 :++
00F2 193 : FUNCTIONAL DESCRIPTION:
00F2 194 :
00F2 195 :--
00F2 196
00F2 197 FREE_UP_MACRO:
00F2 198 PUSHL R6 ;SAVE R6
00F4 199 MOVL MNB$LINK(R1),MNB$LINK(R2) ; Unlink MNB from the list
00F7 200 MOVL R1,R6 ;SAVE POINTER
00FA 201 MOVL W*MAC$GL_INPUTP,R3 ;POINT TO CURRENT INPUT BLOCK
00FF 202 10$: CMPL R1,INP$RPTCNT(R3) ;REDEFINING THIS MACRO?
0103 203 BNEQ 20$ ;IF NEQ NO
0105 204 BBCL #31,INP$RPTCNT(R3),30$ ;YES--FLAG TO DELETE THE OLD DEF.
010A 205 BRB 30$ ;(SAFETY FIRST!)
010C 206 20$: MOVL (R3),R3 ;LINK TO NEXT INPUT BLOCK
010F 207 CMPL R3,#MAC$GL_PRMINBL ;LOOKING AT PRIMARY INPUT BLOCK?
0116 208 BNEQ 10$ ;IF NEQ NO
0118 209 BSBB MAC$DEL_MAC_DEF ;DELETE THE MACRO DEFINITION
011A 210 30$: POPL R6 ;RESTORE R6
011D 211 RSB
011E 212
011E 213 :++
011E 214 : FUNCTIONAL DESCRIPTION:
011E 215 :
011E 216 : THIS ROUTINE DELETES THE MACRO DEFINITION POINTED TO BY R6.
011E 217 : THE MNB IS NOT DEALLOCATED IF CREFFING MACROS.
011E 218 :
011E 219 :
011E 220
011E 221 MAC$DEL_MAC_DEF::
011E 222 DECI W*MAC$GL_MCDEF ;UN-COUNT THE MACRO DEFINITION
0122 223 CLRL MNB$TXTP(R6) ;CLEAR TEXT POINTER
0125 224 BBS #CRF$V MACROS,W*MAC$GL_CRF FLG,5$ ;BRANCH IF CREFFING MACROS
012B 225 MOVZBL MNB$B_NAME(R6),R0 ; Get offset to symbol count/name
012F 226 SUBL3 R0,R6,R0 ; and get start address of page
0133 227 MOVL MNB$PAGP(R6),R6 ;POINT TO THE PAGE LIST
0137 228 BSBW MAC$DEA 1 PAGE ; Deallocate MNB which is always 1 page
013A 229 DECL W*MAC$GL_MCPGRQ ; One less page for macro definitions
013E 230 BRB 20$ ;GO FREE SOME PAGES
0140 231 5$: MOVL MNB$PAGP(R6),R0 ;POINT TO POSSIBLE PAGE TO FREE
0144 232 CLRL MNB$PAGP(R6) ;AND PAGE POINTER
0147 233 MOVL R0,R6 ;ANY PAGES TO FREE?
014A 234 BEQL 30$ ;IF EQL NO--ALL DONE
014C 235 10$: MOVL R6,R0 ;GET SET TO FREE CURRENT PAGE
014F 236 MOVL MNB$LINK(R6),R6 ; Point R6 to possible next block
0152 237 SUBL2 MNB$PAGES(R0), - ; Subtract size of block from pages
0158 238 W*MAC$GL_MCPGRQ ; used for macro definitions
015B 239 BSBW MAC$DEAL_BLOCK ; Deallocate block of memory
015B 240 20$:
015B 241 TSTL R6 ;ANY MORE PAGES?
015D 242 BNEQ 10$ ;IF NEQ YES--GO DO THEM
015F 243 30$: RSB
```

```
0160 245 .SBTTL MACRO_DEFINE CREATE MNB AND SET IT UP
0160 246
0160 247 :++
0160 248 : FUNCTIONAL DESCRIPTION:
0160 249 :
0160 250 : THIS ROUTINE ALLOCATES A PAGE FOR A MACRO NAME BLOCK AND
0160 251 : MACRO ARG BLOCKS. THE MNB IS INITIALIZED AND THE POINTER
0160 252 : IS RETURNED IN MAC$GL_MACPTR.
0160 253 :
0160 254 :--
0160 255
0160 256 MACRO_DEFINE:
0160 257   PUSH R2 ;SAVE PTR TO PREVIOUS
0162 258   MAC$ALL 1 PAGE ;ALLOCATE A PAGE FOR THE MNB
0163 259   INCL W^MAC$GC MCPGRQ ;COUNT PAGE FOR MACRO DEFINITION
0169 260   MOVL R0,W^MAC$GL_BLKPTR ; Set pointer to start of block
016E 261   MOVZBL W^MAC$AB_TMPSYM,R1 ; Get symbol count
0173 262   CMPB R1,#SYM$K_TWOCOL ; Wide listing format required?
0176 263   BLSS 10$ ; No if LSS
0178 264   BBSS #FLG$V_MAC2COL,(R11),10$ ; Set flag for wide macro name listing
017C 265 10$:
017C 266   INCL R1 ; Include the count byte
017E 267   MOVC3 R1,W^MAC$AB_TMPSYM,(R0) ; Copy symbol count/name into MNB
0184 268   MOVL R3,R6 ; Save address of MNB
0187 269   MOVL R6,W^MAC$GL_MACPTR ; and return address in MACPTR
018C 270   POPL R0 ;RESTORE PTR TO PREVIOUS
018F 271   MOVL MNB$LINK(R0),(R3)+ ; Link in new MNB
0192 272   MOVL R6,MNB$LINK(R0)
0195 273   ADDB3 #1,W^MAC$AB_TMPSYM,(R3)+ ; Insert name offset
019B 274   CLRL (R3)+ ;CLEAR TXTP
019D 275   CLRW (R3)+ ;CLEAR FLAGS
019F 276   CLRL (R3)+ ;CLEAR PAGP
01A1 277   MOVL #1,(R3)+ ;INITIALIZE PAGC (PAGE COUNT)
01A4 278   CLRL (R3)+ ;CLEAR CRSYM
01A6 279   CLRB (R3)+ ;CLEAR ARG COUNT
01A8 280   CLRL (R3)+ ;CLEAR ARG POINTER
01AA 281   MOVL R3,W^MAC$GL_ARGPTR ;STORE POINTER FOR MAB ALLOCATION
01AF 282   RSB
```

52 DD 0160 257  
FE9B' 30 0162 258  
0000'CF 50 D0 0163 259  
51 0000'CF 9A 0169 260  
10 51 91 016E 261  
04 19 0173 262  
00 68 2B E2 0176 263  
51 D6 0178 264  
60 0000'CF 51 28 017C 265  
56 53 D0 017E 266  
0000'CF 56 D0 0184 267  
50 8ED0 0187 268  
83 60 D0 018C 269  
60 56 D0 018F 270  
83 0000'CF 01 81 0192 271  
83 83 D4 0195 272  
83 B4 019B 273  
83 D4 019D 274  
83 D4 019F 275  
83 01 D0 01A1 276  
83 D4 01A4 277  
83 94 01A6 278  
0000'CF 83 D4 01A8 279  
53 D0 01AA 280  
05 01AF 281



```
01B0 284 .SBTTL MACRO_ARG_SCAN SCAN FORMAL MACRO ARGUMENTS
01B0 285
01B0 286 :++
01B0 287 : FUNCTIONAL DESCRIPTION:
01B0 288
01B0 289 THIS ROUTINE SCANS THE FORMAL MACRO ARGUMENTS AND STORES
01B0 290 THEM IN MACRO ARGUMENT BLOCKS (MAB'S) THAT ARE LINKED TO
01B0 291 THE MACRO NAME BLOCK.
01B0 292 :--
01B0 293
01B0 294
01B0 295 MACRO_ARG_SCAN:
01B0 296 POSHR #M<R7,R8,R12> ;SAVE REGISTERS
58 1180 8F BB 01B4 297 MOVL W^MAC$GL_ARGPTR,R8 ;POINT TO FREE SPACE IN BLOCK
50 0000'CF D0 01B9 298 MOVZBL MNB$B_NAME(R6),R0 ;Get bytes used to store symbol name
57 58 56 C3 01BD 299 SUBL3 R6,R8,R7 ;Figure bytes used for fixed-sized MNB
57 57 50 C0 01C1 300 ADDL2 R0,R7 ;Add fixed and variable parts
57 00000200 8F 57 C3 01C4 301 SUBL3 R7,#512,R7 ;REALLY GET SPACE LEFT NOW
FE31' 30 01CC 302 BSBW MAC$SKIPSP ;SKIP SPACES
OD 5A 91 01CF 303 CMPB R10,#CR ;SKIP TO END OF LINE?
03 12 01D2 304 BNEQ 10$ ;IF NEQ NO
0150 31 01D4 305 BRW ARG_SCAN_EXIT ;YES--NO FORMAL ARGS HERE
01D7 306 10$:
01D7 307
01D7 308 ARG_SCAN_LOOP:
01D7 309 BSBW MAC$SKIPSP ;SKIP LEADING SPACES
3F 5A 91 01DA 310 CMPB R10,#^A/?/ ;CREATED SYMBOL?
15 12 01DD 311 BNEQ 30$ ;IF NEQ NO
1F 17 A6 91 01DF 312 CMPB MNB$B_ARGCT(R6),#31 ;ONLY FIRST 32 SYMBOLS CAN BE CREATED
OF 1A 01E3 313 BGTRU 30$ ;IF GTR THEN LOSE
50 17 A6 9A 01E5 314 MOVZBL MNB$B_ARGCT(R6),R0 ;GET ARGUMENT NUMBER - 1
OD 13 A6 50 E3 01E9 315 BBCS R0,MNB$B_CRSYMF(R6),20$ ;SET CREATED SYMBOL FLAG FOR THIS ARG
FE0F' 30 01EE 316 20$: BSBW MAC$GETCHR ;SKIP THE QUESTION MARK
FE0C' 30 01F1 317 BSBW MAC$SKIPSP ;SKIP LEADING SPACES
02 E0 01F4 318 30$: BBS #CHR$V_SYM_CHR- ;BRANCH IF LEGAL CHAR FOR ARG
2F 0000'CA 01F6 319 W^MAC$AB_CMSK_TAB(R10),80$
FDFE' 30 01FA 320 $MAC_ERR_ILLMACARGN ;No--get error code
0202 321 BSBW MAC$ERRORLN ;ISSUE ERROR TO PASS 2
0202 322 :
0202 323 : LOOP LOOKING FOR END OF ARGUMENT OR END OF LINE
0202 324
0202 325 40$: BSBW MAC$GETCHR ;GET NEXT CHARACTER
2C 5A 91 0205 326 CMPB R10,#^A/,/ ;START OF NEXT ARG?
11 13 0208 327 BEQL 50$ ;IF EQL YES
09 5A 91 020A 328 CMPB R10,#TAB ;TAB?
OF 13 020D 329 BEQL 60$ ;IF EQL YES
20 5A 91 020F 330 CMPB R10,#^A/ / ;SPACE?
0A 13 0212 331 BEQL 60$ ;IF EQL YES
OD 5A 91 0214 332 CMPB R10,#CR ;NO--END OF LINE?
OD 13 0217 333 BEQL 70$ ;IF EQL YES
E7 11 0219 334 BRB 40$ ;NO--KEEP SKIPPING
021B 335 :
021B 336 : FOUND A COMMA
021B 337
021B 338 50$: BSBW MAC$GETCHR ;SKIP THE COMMA
OD FDE2' 30 021B 339 60$: BSBW MAC$SKIPSP ;SKIP SPACES
FDDF' 30 021E 340 CMPB R10,#CR ;DID WE SKIP TO END OF LINE?
OD 5A 91 0221
```

```

      B1 12 0224 341 BNEQ ARG_SCAN_LOOP ;IF NEQ NO--CONTINUE SCANNING ARGS
      00F9 31 0226 342 70$: BRW 150$ ;QUIT ON EOL
      0229 343
      0229 344 : CHARACTER CAN LEGALLY START A MACRO ARGUMENT
      0229 345
      FDD4' 30 0229 346 80$: BSBW MAC$SYMSCNUP ;GET SYMBOL NAME
      56 56 DD 022C 347 PUSHL R6 ;SAVE PTR TO MNB
      56 18 A6 9E 022E 348 MOVAB MNB$$_ARGP(R6),R6 ;POINT TO ARGUMENT LIST
      55 56 DO 0232 349 MOVL R6,R5 ;MAKE A COPY
      FDC8' 30 0235 350 BSBW MAC$SRC_LIST ;LOOK IT UP
      56 8ED0 0238 351 POPL R6 ;RESTORE MNB POINTER
      50 50 DD 023B 352 PUSHL R0 ;SAVE THE SEARCH RESULT
      0000'CF 51 DO 023D 353 MOVL R1,W^MAC$GL_ARGPTR ;SAVE POINTER TO MAB IN CASE FOUND
      53 06 A1 DE 0242 354 MOVAL MAB$$_DVLENT(R1),R3 ;ASSUME WE FOUND IT
      63 50 E8 0246 355 BLBS R0,100$ ;BRANCH IF FOUND
      53 58 DO 0249 356 MOVL R8,R3 ;NO--GET PTR TO FREE MAB SPOT ON PAGE
      50 0000'CF 9A 024C 357 MOVZBL W^MAC$AB_TMP$SYMBOL,R0 ;Get symbol name length
      50 50 D6 0251 358 INCL R0 ;Include count byte
      57 50 C2 0253 359 SUBL2 R0,R7 ;Subtract from space in block
      57 0C C2 0256 360 SUBL2 #MAB$$_BLKSIZ,R7 ;IS THERE ROOM IN BLOCK FOR MAB?
      25 14 0259 361 BGTR 90$ ;IF GTR YES
      FDA2' 30 025B 362 BSBW MAC$ALL_1_PAGE ;NO--GET ANOTHER PAGE
      0000'CF 50 D6 025E 363 INCL W^MAC$GL_MCPGRQ ;COUNT PAGE FOR MACRO DEFINITION
      58 50 DO 0262 364 MOVL R0,R8 ;PAGE ADDRESS TO R8
      0000'CF 50 DO 0265 365 MOVL R0,W^MAC$GL_BLKPTR ;SET NEW PAGE START
      026A 366 ASSUME MNB$$_LINK EQ 0
      026A 367 ASSUME MNB$$_PAGES EQ MNB$$_LINK+4
      88 0B A6 DO 026A 368 MOVL MNB$$_PAGE(R6),(R8)+ ;LINK NEW PAGE INTO PAGE LIST
      0B A6 50 DO 026E 369 MOVL R0,MNB$$_PAGE(R6)
      88 01 DO 0272 370 MOVL #1,(R8)+ ;Size of block is 1 page
      OF A6 D6 0275 371 INCL MNB$$_PAGE(R6) ;COUNT ANOTHER PAGE FOR DEF.
      57 01EC 8F 3C 0278 372 MOVZWL #<512-MAB$$_BLKSIZ-MAB$$_BLKSIZ>,R7 ;Set block count
      53 58 DO 027D 373 MOVL R8,R3 ;SET POINTER IN R3 ALSO
      0280 374 90$:
      50 0000'CF 52 DD 0280 375 PUSHL R2 ;Save previous pointer
      50 9A 0282 376 MOVZBL W^MAC$AB_TMP$SYMBOL,R0 ;Get symbol name length
      63 0000'CF 50 D6 0287 377 INCL R0 ;Include count byte
      0000'CF 53 DO 0289 378 MOVCL R0,W^MAC$AB_TMP$SYMBOL,(R3) ;Copy symbol count/name to MAB
      52 8ED0 028F 379 MOVL R3,W^MAC$GL_ARGPTR ;Set new argument pointer
      83 62 DO 0294 380 POPL R2 ;Restore previous pointer
      0000'CF 01 DO 0297 381 MOVL MAB$$_LINK(R2),(R3)+ ;Copy previous link to new MAB
      83 0000'CF 81 DO 029A 382 MOVL W^MAC$GL_ARGPTR,MAB$$_LINK(R2) ;Point previous link to new MAB
      17 A6 96 029F 383 ADDB3 #1,W^MAC$AB_TMP$SYMBOL,(R3)+ ;Store name offset
      83 17 A6 90 02A5 384 INCB MNB$$_ARGCT(R6) ;COUNT THE ARGUMENT
      5C 53 DO 02A8 385 MOVAB MNB$$_ARGCT(R6),(R3)+ ;STORE ARGUMENT NUMBER
      83 B4 02AC 386 100$: MOVL R3,R12 ;SAVE KEYWORD LINK POINTER
      83 D4 02AF 387 CLRW (R3)+ ;CLEAR DEFAULT VALUE LENGTH
      8E D5 02B1 388 CLRL (R3)+ ;CLEAR DEFAULT VALUE PTR
      03 12 02B3 389 TSTL (SP)+ ;DID WE FIND THE MAB OR MAKE NEW ONE?
      58 53 DO 02B5 390 BNEQ 110$ ;IF NEQ WE FOUND IT--DON'T UPDATE PTR
      FD43' 30 02B7 391 MOVL R3,R8 ;STORE POINTER TO NEXT MAB
      3D 5A 91 02BA 392 110$: BSBW MAC$SKIPSP ;SKIP SPACES
      50 12 02BD 393 CMPB R10,#^A/=/ ;KEYWORD ARG?
      02C0 394 BNEQ 130$ ;IF NEQ NO
      02C2 395
      02C2 396 : SCAN DEFAULT ARGUMENT
      02C2 397
```

```
FD3B' 30 02C2 398 BSBW MAC$GETCHR ;GET FIRST CHARACTER OF ARGUMENT
FD3B' 30 02C3 399 BSBW MAC$MAC_ARG_SCN ;SCAN DEFAULT ARG
50 D5 02C8 400 TSTL R0 ;WAS IT NULL?
46 13 02CA 401 BEQL 130$ ;IF EQL YES
8C 50 B0 02CC 402 MOVW R0,(R12)+ ;STORE DEFAULT VALUE LENGTH
57 50 C2 02CF 403 SUBL2 R0,R7 ;IS THERE ROOM FOR DEFAULT STRING?
32 14 02D2 404 BGTR 120$ ;IF GTR YES
50 DD 02D4 405 PUSHL R0 ;NO--SAVE SIZE OF DEFAULT ARG
51 50 08 C1 02D6 406 ADDL3 #MXB$K_BLKSI2,R0,R1 ; Compute total size of block required
FD23' 30 02DA 407 BSBW MAC$ALC_BLOCK ; Allocate block of memory
0000'CF 51 C0 02DD 408 ADDL2 R1,W*MAC$GL_MCPGRQ ; Add pages allocated to macro definitions
58 50 D0 02E2 409 MOVL R0,R8 ;PAGE ADDRESS TO R8
0000'CF 50 D0 02E3 410 MOVL R0,W*MAC$GL_BLKPTR ;SET PAGE START ADDRESS IN BLKPTR
88 0B A6 D0 02EA 411 MOVL MNBSL_PAGP(R6),(R8)+ ;LINK NEW PAGE INTO PAGE LIST
0B A6 50 D0 02EE 412 MOVL R0,MNBSL_PAGP(R6)
88 51 D0 02F2 413 MOVL R1,(R8)+ ; Insert block size in block
OF A6 51 C0 02F3 414 ADDL2 R1,MNBSL_PAGC(R6) ; Add pages to total used to define macro
57 51 09 78 02F9 415 ASHL #9,R1,R7 ; Compute block size in bytes
57 08 C2 02FD 416 SUBL2 #MXB$K_BLKSI2,R7 ; and subtract MXB header size
50 8ED0 0300 417 POPL R0 ;SAVE SIZE OF DEFAULT ARG
57 50 C2 0303 418 SUBL2 R0,R7 ; Subtract size of argument
8C 58 D0 0306 419 120$: MOVL R8,(R12)+ ;STORE DEFAULT VALUE POINTER
68 0000'CF 50 28 0309 420 MOVCL3 R0,W*MAC$AB_TMPBUF,(R8) ;COPY INTO BLOCK
58 53 D0 030F 421 MOVL R3,R8 ;UPDATE ARG POINTER
2C 03 91 0312 422 130$: CMPB R10,#^A/,/ ;STOP ON A COMMA?
03 12 0315 423 BNEQ 140$ ;IF NEQ NO
FCE6' 30 0317 424 BSBW MAC$GETCHR ;YES--GET NEXT CHARACTER
OD 5A 91 031A 425 140$: CMPB R10,#CR ;END OF LINE?
03 13 031D 426 BEQL 150$ ;IF EQL NO
FEB5' 31 031F 427 BRW ARG_SCAN_LOOP ;NO--CONTINUE SCANNING ARGS
0000'CF 58 D0 0322 428 150$: MOVL R8,W*MAC$GL_ARGPTR ;STORE UPDATED ARG POINTER
0327 429
0327 430 ARG_SCAN_EXIT:
1180 8F BA 0327 431 POPR #^M<R7,R8,R12> ;RESTORE REGISTERS
05 032B 432 RSB
032C 433
032C 434 .END
```



MACSMACDEF  
Symbol table

## MACRO DEFINITION PROCESSOR

M 6

16-SEP-1984 02:07:14  
5-SEP-1984 01:48:54VAX/VMS Macro V04-00  
[MACRO.SRC]MACDEF.MAR;1Page 12  
(7)

SCOUNT = 0000003B  
ARGSK\_SIZE = 000003E8  
ARG\_SCAN\_EXIT = 00000327 R 03  
ARG\_SCAN\_LOOP = 000001D7 R 03  
AUDSK\_SIZE = 00000010  
BLNK = 00000020  
CHRSM\_COMMA CR = 00000020  
CHRSM\_ILL\_CHR = 00000040  
CHRSM\_NUM\_BER = 00000010  
CHRSM\_SPA\_MSK = 00000001  
CHRSM\_SYM\_CH1 = 00000008  
CHRSM\_SYM\_CHR = 00000004  
CHRSM\_SYM\_DLM = 00000002  
CHRSM\_COMMA CR = 00000005  
CHRSM\_CVTLWC = 00000061  
CHRSM\_ILL\_CHR = 00000006  
CHRSM\_NOCVT = 0000007F  
CHRSM\_NUM\_BER = 00000004  
CHRSM\_SPA\_MSK = 00000000  
CHRSM\_SYM\_CH1 = 00000003  
CHRSM\_SYM\_CHR = 00000002  
CHRSM\_SYM\_DLM = 00000001  
CNT = 00000001  
CR = 0000000D  
CRFSK\_DEF = \*\*\*\*\* X 03  
CRFSM\_DEFAULT = 00000012  
CRFSM\_DIR = 00000001  
CRFSM\_MACROS = 00000002  
CRFSM\_OPCODES = 00000004  
CRFSM\_REGISTERS = 00000008  
CRFSM\_SYMBOLS = 00000010  
CRFSV\_DIR = 00000000  
CRFSV\_MACROS = 00000001  
CRFSV\_OPCODES = 00000002  
CRFSV\_REGISTERS = 00000003  
CRFSV\_SYMBOLS = 00000004  
DELETE = 000000AF RG 03  
ERR = 00000000  
FF = 0000000C  
FLGSM\_ALLCHR = 00000001  
FLGSM\_BOL = 00000002  
FLGSM\_CHKLPND = 00100000  
FLGSM\_COMPEXPR = 00000004  
FLGSM\_CONT = 00000008  
FLGSM\_CRF = 40000000  
FLGSM\_CRSEEN = 00000001  
FLGSM\_DATRPT = 00000010  
FLGSM\_DBGOUT = 00004000  
FLGSM\_DLMSTR = 00008000  
FLGSM\_ENDMCH = 00000020  
FLGSM\_EVALEXPR = 00000040  
FLGSM\_EXPOPT = 00000080  
FLGSM\_EXTERR = 00010000  
FLGSM\_EXTWRN = 00020000  
FLGSM\_FIRSTLN = 00000200  
FLGSM\_IFSTAT = 00800000  
FLGSM\_IIF = 00400000

FLGSM\_INSERT = 00000100  
FLGSM\_IRPC = 20000000  
FLGSM\_LEXOP = 00000002  
FLGSM\_LSTXST = 00000200  
FLGSM\_MAC2COL = 00000800  
FLGSM\_MACL = 00000800  
FLGSM\_MACLTB = 08000000  
FLGSM\_MACTXT = 00010000  
FLGSM\_MEBLST = 00001000  
FLGSM\_MOREARG = 00002000  
FLGSM\_MOREINP = 00000008  
FLGSM\_NEWPND = 00000400  
FLGSM\_NOREF = 01000000  
FLGSM\_NTTYPEPC = 00000020  
FLGSM\_NULCHR = 00040000  
FLGSM\_OBJXST = 00200000  
FLGSM\_OPNDCHK = 00000100  
FLGSM\_OPRND = 00002000  
FLGSM\_OPTVFLIDX = 00001000  
FLGSM\_ORDLST = 00020000  
FLGSM\_P2 = 00004000  
FLGSM\_RPTIRP = 10000000  
FLGSM\_SEQFIL = 02000000  
FLGSM\_SKAN = 00008000  
FLGSM\_SPECOP = 00000004  
FLGSM\_SPLALL = 04000000  
FLGSM\_STOIMF = 00040000  
FLGSM\_SYM2COL = 00000400  
FLGSM\_TOCLFG = 00080000  
FLGSM\_UPAFLG = 00000010  
FLGSM\_UPDFIL = 00000080  
FLGSM\_UPMARG = 00000040  
FLGSM\_XCRF = 80000000  
FLGSM\_ALLCHR = 00000000  
FLGSM\_BOL = 00000001  
FLGSM\_CHKLPND = 00000014  
FLGSM\_COMPEXPR = 00000002  
FLGSM\_CONT = 00000003  
FLGSM\_CRF = 0000001E  
FLGSM\_CRSEEN = 00000020  
FLGSM\_DATRPT = 00000004  
FLGSM\_DBGOUT = 0000002E  
FLGSM\_DLMSTR = 0000002F  
FLGSM\_ENDMCH = 00000005  
FLGSM\_EVALEXPR = 00000006  
FLGSM\_EXPOPT = 00000007  
FLGSM\_EXTERR = 00000030  
FLGSM\_EXTWRN = 00000031  
FLGSM\_FIRSTLN = 000C0029  
FLGSM\_IFSTAT = 00000017  
FLGSM\_IIF = 00000016  
FLGSM\_INSERT = 00000008  
FLGSM\_IRPC = 0000001D  
FLGSM\_LEXOP = 00000021  
FLGSM\_LSTXST = 00000009  
FLGSM\_MAC2COL = 0000002B  
FLGSM\_MACL = 0000000B

FLGSM\_MACLTB = 0000001B  
FLGSM\_MACTXT = 00000010  
FLGSM\_MEBLST = 0000000C  
FLGSM\_MOREARG = 0000002D  
FLGSM\_MOREINP = 00000023  
FLGSM\_NEWPND = 0000000A  
FLGSM\_NOREF = 00000018  
FLGSM\_NTTYPEPC = 00000025  
FLGSM\_NULCHR = 00000032  
FLGSM\_OBJXST = 00000015  
FLGSM\_OPNDCHK = 00000028  
FLGSM\_OPRND = 0000000D  
FLGSM\_OPTVFLIDX = 0000002C  
FLGSM\_ORDLST = 00000011  
FLGSM\_P2 = 0000000E  
FLGSM\_RPTIRP = 0000001C  
FLGSM\_SEQFIL = 00000019  
FLGSM\_SKAN = 0000000F  
FLGSM\_SPECOP = 00000022  
FLGSM\_SPLALL = 0000001A  
FLGSM\_STOIMF = 00000012  
FLGSM\_SYM2COL = 0000002A  
FLGSM\_TOCLFG = 00000013  
FLGSM\_UPAFLG = 00000024  
FLGSM\_UPDFIL = 00000027  
FLGSM\_UPMARG = 00000026  
FLGSM\_XCRF = 0000001F  
FREE\_OP\_MACRO = 000000F2 R 03  
HASHSZ = 0000007F  
HYPHEN = 0000002D  
INPSB\_ARGCT = 0000001C  
INPSK\_BLKSI2 = 00000021  
INPSK\_BUFSI2 = 000003E8  
INPSK\_IRPSI2 = 0000003C  
INPSL\_ARGS = 0000001D  
INPSL\_GETL = 00000008  
INPSL\_IFLVL = 0000000C  
INPSL\_IFVAL = 00000010  
INPSL\_LINK = 00000000  
INPSL\_NXTL = 00000004  
INPSL\_PAGP = 00000018  
INPSL\_RPTCNT = 00000014  
INTSK\_BUFSI2 = 000013F4  
INTSK\_BUFWRN = 00001390  
INTS\_ADD = 00000001  
INTS\_AND = 00000002  
INTS\_ASH = 00000003  
INTS\_ASN = 0000000C  
INTS\_AUGPC = 0000000D  
INTS\_BDST = 0000000E  
INTS\_CHKL = 0000000F  
INTS\_DIV = 00000004  
INTS\_END = 00000010  
INTS\_EPT = 00000011  
INTS\_ERR = 00000012  
INTS\_ETX = 00000013  
INTS\_FNEWL = 00000014

MAC\$MACDEF  
Symbol table

## MACRO DEFINITION PROCESSOR

N 6

16-SEP-1984 02:07:14  
5-SEP-1984 01:48:54VAX/VMS Macro V04-00  
[MACRO.SRC]MACDEF.MAR;1Page 13  
(7)

```
INT$_ILG      = 00000000
INT$_INFO     = 0000003A
INT$_LGLAB    = 00000015
INT$_MACL     = 00000016
INT$_MUL      = 00000005
INT$_NEG      = 00000006
INT$_NEWL     = 00000017
INT$_NEWP     = 00000018
INT$_NOT      = 00000007
INT$_OP       = 00000019
INT$_OR       = 00000008
INT$_PRIL     = 0000001A
INT$_PRT      = 0000001B
INT$_PSECT    = 0000001C
INT$_REDEF    = 0000001D
INT$_REF      = 0000001E
INT$_REST     = 0000001F
INT$_SAME     = 00000009
INT$_SAVE     = 00000020
INT$_SBTTL    = 00000021
INT$_SETFLAG  = 00000022
INT$_SETLONG  = 00000023
INT$_SPIC     = 00000024
INT$_SPID     = 00000025
INT$_STIB     = 00000026
INT$_STIL     = 00000028
INT$_STIW     = 00000027
INT$_STKEPT   = 00000029
INT$_STKG     = 0000002A
INT$_STKL     = 0000002B
INT$_STKPC    = 0000002C
INT$_STKS     = 0000002D
INT$_STOB     = 00000034
INT$_STOL     = 0000002E
INT$_STOW     = 00000035
INT$_STRB     = 0000002F
INT$_STRL     = 00000031
INT$_STRSB    = 00000032
INT$_STRSW    = 00000033
INT$_STRW     = 00000030
INT$_STSB     = 00000036
INT$_STSW     = 00000037
INT$_SUB      = 0000000A
INT$_SUME     = 00000039
INT$_WRH      = 00000038
INT$_XOR      = 0000000B
LST$G_MACRODEF ***** X 03
LST$K_BUFSIZ  = 00000086
LST$K_L_P_PAGE = 0000003C
LST$K_TITLE_SIZE = 00000028
MAB$B_ARGNO   = 00000005
MAB$B_NAME    = 00000004
MAB$K_BLKSI2  = 0000000C
MAB$B_DVPTTR  = 00000008
MAB$B_LINK    = 00000000
MAB$B_DVLEN   = 00000006
MAC$AB_CMSK_TAB ***** X 03
```

```
MAC$AB_TMPBUF ***** X 03
MAC$AB_TMPSYM ***** X 03
MAC$ALC_1_PAGE ***** X 03
MAC$ALL_BLOCK ***** X 03
MAC$AL_OMCHSHTB ***** X 03
MAC$BODY_SCAN ***** X 03
MAC$CREF_MACRO ***** X 03
MAC$DEAL_BLOCK ***** X 03
MAC$DEAL_T_PAGE ***** X 03
MAC$DEL_MAC_DEF 0000011E RG 03
MAC$ERRORLN ***** X 03
MAC$ERRORPT ***** X 03
MAC$GETCHR ***** X 03
MAC$GL_ARGPTR ***** X 03
MAC$GL_BLKPTR ***** X 03
MAC$GL_CRF_FLG ***** X 03
MAC$GL_INPTP ***** X 03
MAC$GL_LIST_IT ***** X 03
MAC$GL_MACPTR ***** X 03
MAC$GL_MDEF ***** X 03
MAC$GL_MCPGRQ ***** X 03
MAC$GL_PMINBL ***** X 03
MAC$INTOUT_1_LW ***** X 03
MAC$INTOUT_2_LW ***** X 03
MAC$INTOUT_X ***** X 03
MAC$MAC_ARG_SCN ***** X 03
MAC$SKIPSP ***** X 03
MAC$SRCSYMTAB ***** X 03
MAC$SRC_LIST ***** X 03
MAC$SYM$CNUP ***** X 03
MAC$_DIRSYNX = 007D906A
MAC$_ENDWRNGMAC = 007D907A
MAC$_ILLMACARGN = 007D90EA
MAC$_ILLMACNAM = 007D90F2
MACRO 00000000 RG 03
MACRO_ARG_SCAN 000001B0 R 03
MACRO_DEFINE 00000160 R 03
MAC_SUBSYS = 0000007D
MNBS$B_ARGCT 00000017
MNBS$B_NAME 00000004
MNBS$K_BLKSI2 0000001C
MNBS$L_ARGP 00000018
MNBS$L_CRSYMF 00000013
MNBS$L_LINK 00000000
MNBS$L_PAGC 0000000F
MNBS$L_PAGP 0000000B
MNBS$L_TXTP 00000005
MNBS$W_FLAG 00000009
MXB$K_BLKSI2 00000008
MXB$L_LINK 00000000
MXB$L_PAGES 00000004
OBJ$K_BUFSIZ = 00000200
OPF$M_LASTOPR = 00002000
OPF$M_OPTEXP = 00001000
OPF$V_LASTOPR = 0000000D
OPF$V_OPTEXP = 0000000C
PSC$B_NAME 00000004
```

```
PSC$B_SEG 0000000C
PSC$B_UNUSED 0000000B
PSC$K_BLKSI2 00000013
PSC$K_NO_OPTNS = 0000000A
PSC$L_CURLOC 0000000F
PSC$L_LINK 00000000
PSC$L_MAXLGTH 00000005
PSC$M_ABS = FFFFFFFF7
PSC$M_ALIGNFLG = 00004000
PSC$M_ALLOPTNS = 000003FF
PSC$M_BYTE = 00004000
PSC$M_CON = FFFFFFFFB
PSC$M_DEFAULT = 000001C8
PSC$M_EXE = 000000C0
PSC$M_GBL = 00000010
PSC$M_LCL = FFFFFFFEF
PSC$M_LIB = 00000002
PSC$M_LONG = 00004800
PSC$M_NOEXE = FFFFFFFBF
PSC$M_NOPIC = FFFFFFFFE
PSC$M_NORD = FFFFFFF7F
PSC$M_NOSHR = FFFFFFFDF
PSC$M_NOVEC = FFFFFFFDF
PSC$M_NOWRT = FFFFFFFEF
PSC$M_OVR = 00000004
PSC$M_PAGE = 00006400
PSC$M_PIC = 00000001
PSC$M_QUAD = 00004C00
PSC$M_RD = 00000080
PSC$M_REL = 00000008
PSC$M_SHR = 00000020
PSC$M_USR = FFFFFFFFD
PSC$M_VEC = 00000200
PSC$M_WORD = 00004400
PSC$M_WRT = 00000180
PSC$S_ALIGNMENT = 00000004
PSC$V_ALIGNFLG = 0000000E
PSC$V_ALIGNMENT = 0000000A
PSC$V_EXE = 00000006
PSC$V_GBL = 00000004
PSC$V_LIB = 00000001
PSC$V_OVR = 00000002
PSC$V_PIC = 00000000
PSC$V_RD = 00000007
PSC$V_REL = 00000003
PSC$V_SHR = 00000005
PSC$V_VEC = 00000009
PSC$V_WRT = 00000008
PSC$W_FLAG 00000009
PSC$W_OPTIONS 0000000D
RDX$V_BINARY = 00000000
RDX$V_DECIMAL = 00000002
RDX$V_DOUBLE = 00000005
RDX$V_FLOAT = 00000004
RDX$V_GFLOAT = 00000006
RDX$V_HEX = 00000003
RDX$V_HFLOAT = 00000007
```



RDX\$V\_OCTAL = 00000001  
REG\$\_PC = 0000000F  
SEMI = 0000003B  
STBSK\_PG\_MISS = 0000000A  
SYMSB\_NAME = 00000004  
SYMSB\_SEG = 0000000C  
SYMSB\_TOKEN = 0000000B  
SYMSK\_BLKSI2 = 0000000D  
SYMSK\_MAXLEN = 0000001F  
SYMSK\_TWOCOL = 00000010  
SYMSL\_LINK = 00000000  
SYMSL\_VAL = 00000005  
SYMSM\_ABS = 00000010  
SYMSM\_ASN = 00000100  
SYMSM\_CRFO = 00002000  
SYMSM\_DEBUG = 00000020  
SYMSM\_DEF = 00000001  
SYMSM\_DELMAC = 00000200  
SYMSM\_EPT = 00000200  
SYMSM\_EXTRN = 00000008  
SYMSM\_GLOBL = 00000004  
SYMSM\_LOCAL = 00000040  
SYMSM\_ODBG = 00000400  
SYMSM\_REF = 00000080  
SYMSM\_RELPSECT = 00000800  
SYMSM\_SUPR = 00004000  
SYMSM\_WEAK = 00000002  
SYMSM\_XCRF = 00001000  
SYMSV\_ABS = 00000004  
SYMSV\_ASN = 00000008  
SYMSV\_CRFO = 0000000D  
SYMSV\_DEBUG = 00000005  
SYMSV\_DEF = 00000000  
SYMSV\_DELMAC = 00000009  
SYMSV\_EPT = 00000009  
SYMSV\_EXTRN = 00000003  
SYMSV\_GLOBL = 00000002  
SYMSV\_LOCAL = 00000006  
SYMSV\_ODBG = 0000000A  
SYMSV\_REF = 00000007  
SYMSV\_RELPSECT = 0000000B  
SYMSV\_SUPR = 0000000E  
SYMSV\_WEAK = 00000001  
SYMSV\_XCRF = 0000000C  
SYMSW\_FLAG = 00000009  
TAB = 00000009  
X1 = 00000033  
X2 = 00080000

+-----+  
! Psect synopsis !  
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS :	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
. BLANK :	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE



MAC\$MACDEF  
Psect synopsis

MACRO DEFINITION PROCESSOR

C 7

16-SEP-1984 02:07:14 VAX/VMS Macro V04-00  
5-SEP-1984 01:48:54 [MACRO.SRC]MACDEF.MAR;1

Page 15  
(7)

\$AB\$\$  
MAC\$RO\_CODE\_MAC

0000003C ( 60.) 02 ( 2.) NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE  
0000032C ( 812.) 03 ( 3.) NOPIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC LONG

+-----+  
! Performance indicators !  
+-----+

Phase	Page faults	CPU Time	Elapsed Time
-----	-----	-----	-----
Initialization	29	00:00:00.05	00:00:00.82
Command processing	117	00:00:00.37	00:00:04.41
Pass 1	214	00:00:03.45	00:00:13.78
Symbol table sort	0	00:00:00.48	00:00:01.66
Pass 2	95	00:00:00.84	00:00:03.45
Symbol table output	32	00:00:00.19	00:00:01.72
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	491	00:00:05.41	00:00:25.87

The working set limit was 1350 pages.  
31798 bytes (63 pages) of virtual memory were used to buffer the intermediate code.  
There were 30 pages of symbol table space allocated to hold 466 non-local and 34 local symbols.  
434 source lines were read in Pass 1, producing 19 object records in Pass 2.  
16 pages of virtual memory were used to define 15 macros.

+-----+  
! Macro library statistics !  
+-----+

Macro library name	Macros defined
-----	-----
_\$255\$DUA28:[MACRO.OBJ]MACRO.MLB;1	13
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	17

586 GETS were required to define 17 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:MACDEF/OBJ=OBJ\$:MACDEF MSRC\$:MACDEF/UPDATE=(ENH\$:MACDEF)+LIB\$:MACRO/LIB



0226 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

